Philip Müller
René Siewert

Algorithmen und Programmierung III

Musterlösung Übung 6

Aufgabe 36:
Wir wollen eine Variante von *a*-*b*-Bäumen konstruieren, bei der weniger Speicher verschwendet wird. Es sei $b = 100$. Wie groß kann man a wählen, wenn man

- beim Entfernen bis zu *zwei* Geschwisterknoten zum Ausborgen in Betracht zieht, und
- beim Einfügen einen Geschwisterknoten zu Hilfe nimmt, falls dieser den Überlauf aufnehmen kann?

Worst-Case Betrachtung:
Entfernt man ein Kind aus einem minimal besetzten Knoten, der zwei minimal besetzte Geschwisterknoten hat, so werden diese drei Knoten zu zweien zusammengefasst, da einer der Knoten weniger Kinder hat als minimal zugelassen. Sei der Minimalwert a. Vor dem Entfernen hatten alle drei Knoten a Kinder. Nach dem Entfernen haben die zwei Knoten insgesamt 3a-1 Kinder. Jeder Knoten hat also $\frac{3a-1}{2}$ Kinder.
Fügt man ein Kind zu einem Knoten hinzu, der bereits maximal besetzt ist (Anzahl der Kinder = 6), dessen Geschwisterknoten ebenfalls alle maximal besetzt sind, so muss ein neuer Knoten 2b+1 Kinder verteilt werden. also fallen pro Knoten $\frac{2b+1}{3}$ Kinder ab.

Daraus folgt: $a \le \frac{2b+1}{3}$

Nach diesen Ungleichungen ergibt sich für b=100: $a \le \frac{200+1}{3} = 67$

Und die andere Ungleichung : $100 \ge \frac{3a-1}{2}$ ist für a=67 auch erfüllt.

Wurzelbetrachtung:
Da am Anfang der Baum erst aufgebaut werden muss, gibt es keine Minimalanzahl an Kindern. Die Maximalanzahl der Kinder muss auf 133 gesetzt werden, damit beim hinzufügen eines zusätzlichen Knotens die Kinder eine Minimalanzahl von 67 haben.

Aufgabe 37:
Zeigen Sie, dass man aus jedem Rot-Schwarz-Baum einen 2-3-Baum machen kann, und umgekehrt.

Rot-Schwarz Baum → 2-3 Baum
Um einen Rot-Schwarz Baum in einen 2-3 Baum zu konvertieren muss man alle Blätter auf die selbe Höhe bringen. Dies erreicht man, indem man sich die Eigenschaft des 2-3 Baums zunutze macht, dass jeder Knoten auch drei Kinder haben darf. Ohne rote Knoten wären alle Blätter im Rot-Schwarz Baum auf gleicher Höhe. Man fügt nun einfach die „überschüssigen“ roten Knoten mit ihren schwarzen Vorgängern (jeder rote Knoten muss einen schwarzen Vorgänger haben) zu einem einzigen Knoten zusammen der drei Nachfolger hat (zwei von dem roten Knoten und einen den er ohnehin schon hatte).

2-3 Baum → Rot-Schwarz Baum
Das Verfahren ist vom Schema her fast das selbe wie oben, nur das hier rückwärts gearbeitet wird. Was wirklich stört sind hier die Knoten mit drei Kindern. Man kann sich dieser entledigen, indem man aus diesen Knoten zwei macht. Einer dieser Knoten ist dabei Kind des anderen. Der Kindknoten wird rot und erhält zwei Nachfolger des Ursprungsknoten als Kinder. Sein Vorgänger wird schwarz und hat neben dem roten den verbleibenden Nachfolger des Ursprungsknoten als Kind. Alle restlichen Knoten werden schwarz.

Aufgabe 38:
Es sei A_h die kleinstmögliche Anzahl von Blättern eines AVL-Baumes mit Höhe *h*. Beweisen Sie, dass die Zahlen A_h die Fibonacci-Zahlen sind.
Was können Sie daraus für die kleinstmögliche Anzahl $n(h)$ von *inneren Knoten* eines AVL-Baumes mit Höhe *h* schließen?

Konstruktiver Beweis:
Ein AVL-Baum der Höhe 0 hat minimal ein Blatt (mit weniger wäre es ein leerer Baum, mit mehr könnte die Höhe nicht mehr stimmen).
Ein AVL-Baum der Höhe 1 hat minimal einen inneren Knoten und ein Blatt. Der linke Teilbaum des inneren Knoten (der Wurzel) hat die Höhe –1 und der Rechte hat die Höhe 0. Dem zufolge erfüllt dieser Baum die AVL-Baum Bedingungen.
Alle weiteren Bäume kann man nun aus den vorigen zusammenbauen, indem man eine Wurzel nimmt und als rechten Teilbaum den Minimalbaum mit der um 1 verringerten Höhe nimmt und als linken Teilbaum den Minimalbaum mit der um 2 verringerten Höhe nimmt. Man könnte auch sagen, dass man den Minimalbaum mit h-1 als linken Teilbaum verwendet. Dieser ist zwar auch für seine Höhe minimal, die AVL-Eigenschaft erlaubt, aber einen Höhenunterschied von 1 zwischen linken und rechten Teilbäumen. Da der Minimalbaum von h-2 „minimaler“ ist als der Minimalbaum von h-1 wählt man diesen um den minimalsten Gesamtbaum zu erzeugen.
Somit kann man sehen, dass der aktuelle Minimalbaum aus seinen beiden Vorgängen besteht, demnach auch die Summe seiner Blätter. Der kleinste und zweitkleinste Baum haben jeweils 1 Blatt, wie auch die beiden Startzahlen der Fibonacci Zahlen. Da sie nach gleichem Schema wachsen kann man sie gleichsetzen.
Die Summe der inneren Knoten eines Minimalbaums der Höhe h entspricht der Summe der inneren Knoten des linken und rechten Teilbaums plus eins, da ein neuer Knoten als Wurzel hinzukommt. Die Rekursionsanker sind hier 0 für den kleinsten Baum und 1 für den zweitkleinsten Baum. Also:
$\#innereKnoten(h) = A_{h+1} - 1$ und damit auch die Fibonacci Zahlen.